

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2014-509012
(P2014-509012A)

(43) 公表日 平成26年4月10日(2014.4.10)

(51) Int.Cl.

G06F 11/14 (2006.01)

F 1

G06F 11/14 310B

テーマコード(参考)

5B027

審査請求 未請求 予備審査請求 有 (全 25 頁)

(21) 出願番号 特願2013-554594 (P2013-554594)
 (86) (22) 出願日 平成24年2月16日 (2012.2.16)
 (85) 翻訳文提出日 平成25年7月18日 (2013.7.18)
 (86) 國際出願番号 PCT/US2012/025388
 (87) 國際公開番号 WO2012/112748
 (87) 國際公開日 平成24年8月23日 (2012.8.23)
 (31) 優先権主張番号 13/030,998
 (32) 優先日 平成23年2月18日 (2011.2.18)
 (33) 優先権主張国 米国(US)

(71) 出願人 509123208
 アビニシオ テクノロジー エルエルシー
 アメリカ合衆国 02421 マサチュー
 セツツ州 レキシントン スプリング ス
 トリート 201
 (74) 代理人 100079108
 弁理士 稲葉 良幸
 (74) 代理人 100109346
 弁理士 大貫 敏史
 (72) 発明者 ドゥラス, ブライアン フィル
 アメリカ合衆国, マサチューセツツ州 1
 0701, フラミングム, レイクビュー
 ロード 92

最終頁に続く

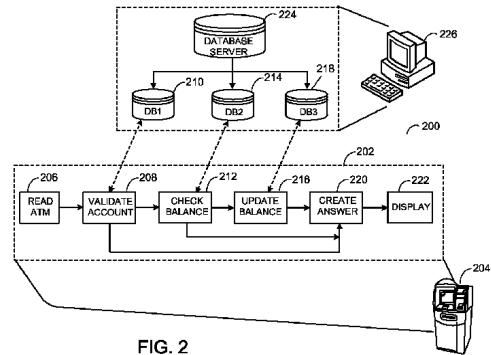
(54) 【発明の名称】プロセスの再開

(57) 【要約】

【課題】

【解決手段】コンピュータにより実施される方法を含む
 技法を開示する。この方法は、初期化の時にプロセスの
 初期状態に関する情報を記憶すること(402)であつて、
 プロセスの実行が少なくとも1つの実行フェーズを
 実行することを含み、及び実行フェーズの実行の完了時
 に実行フェーズの終了状態を表す情報を記憶すること(404)
 と、所定のイベントに応答して前記プロセスの
 実行をアボートすること(506)と、プロセスをシャ
 ットダウンする必要なく、保存された初期状態及び終了
 状態の一方からプロセスの実行を再開すること(512)
 と、を含む。

【選択図】図2



【特許請求の範囲】**【請求項 1】**

コンピュータにより実施される方法であって、

初期化の時にプロセスの初期状態に関する情報を記憶することであって、前記プロセスの実行が少なくとも 1 つの実行フェーズを実行することを含み、及び前記実行フェーズの前記実行の完了時に前記実行フェーズの終了状態を表す情報を記憶することと、

所定のイベントに応答して前記プロセスの実行をアボートすることと、

前記プロセスをシャットダウンする必要なく、前記保存された初期状態及び終了状態の一方から前記プロセスの実行を再開することと、

を含む、方法。

10

【請求項 2】

前記所定のイベントが外部デバイスに対する接続の切断を表す、請求項 1 に記載のコンピュータにより実施される方法。

【請求項 3】

前記所定のイベントが外部デバイスによるエラーを表す、請求項 1 に記載のコンピュータにより実施される方法。

【請求項 4】

前記外部デバイスに対する前記接続が復元された場合に、前記プロセスの前記実行が再開される、請求項 2 に記載のコンピュータにより実施される方法。

【請求項 5】

前記外部デバイスによる前記エラーがクリアされた場合に、前記プロセスの前記実行が再開される、請求項 3 に記載のコンピュータにより実施される方法。

20

【請求項 6】

前記所定のイベントが発生した実行フェーズの前に記憶された終了状態から、前記プロセスの前記実行が再開される、請求項 1 に記載のコンピュータにより実施される方法。

【請求項 7】

前記プロセスの実質的に起動直後の実行フェーズ中に前記所定のイベントが発生した場合、前記初期状態から前記プロセスの前記実行を再開する、請求項 1 に記載のコンピュータにより実施される方法。

【請求項 8】

実行フェーズの実行が、受信したデータストリームに 1 つ以上の処理アクションを実行して前記実行フェーズに対応する出力データを生成することを含む、請求項 1 に記載のコンピュータにより実施される方法。

30

【請求項 9】

1 つ以上の実行フェーズに対応した前記出力データを記憶することと、
前記プロセスの実行が再開される場合に前記出力データを再現することと、
を更に含む、請求項 8 に記載のコンピュータにより実施される方法。

【請求項 10】

前記プロセスが、プロセスステージの一部であり、データ経路を介して前記プロセスステージにおける第 2 の異なるプロセスと通信状態にある、請求項 1 に記載のコンピュータにより実施される方法。

40

【請求項 11】

前記プロセスステージにおける各プロセスを介してチェックポイントコマンドメッセージを伝達することと、

各プロセスにおいて、前記チェックポイントコマンドメッセージの受信時に初期状態又は終了状態に関する新しい情報を保存することであって、前記保存することが、前記プロセスの動作を中断すること及び前記新しい情報を記憶領域に保存することを含む、ことと、

を更に含む、請求項 10 に記載のコンピュータにより実施される方法。

【請求項 12】

50

古い保存された初期状態又は終了状態を前記新しい初期状態又は終了状態で上書きすることを更に含む、請求項11に記載のコンピュータにより実施される方法。

【請求項13】

各プロセスが、前記プロセスのためのデータを受信しキューに入れるためのデータキューと通信状態にある、請求項11に記載のコンピュータにより実施される方法。

【請求項14】

トリガイベントの検出に応答して前記チェックポイントコマンドメッセージを生成することを更に含む、請求項11に記載のコンピュータにより実施される方法。

【請求項15】

前記トリガイベントがネットワークイベントに関する情報を含む、請求項14に記載のコンピュータにより実施される方法。 10

【請求項16】

前記チェックポイントコマンドメッセージを定期的に生成することを更に含む、請求項11に記載のコンピュータにより実施される方法。

【請求項17】

処理している受信データレコード内の又は前記データレコードから導出される選択されたデータ値の発生に応答して前記チェックポイントコマンドメッセージを生成することを更に含む、請求項11に記載のコンピュータにより実施される方法。

【請求項18】

前記所定のイベントに応答してアボートコマンドメッセージを生成することと、
前記プロセスステージにおける各プロセスを介して前記アボートコマンドメッセージを伝達することと、 20

各プロセスにおける前記アボートコマンドメッセージの受信時に、前記プロセスの実行をアボートし、前記アボートコマンドメッセージを次のプロセスに伝達することと、
を更に含む、請求項11に記載のコンピュータにより実施される方法。

【請求項19】

処理再開メッセージに含まれた情報に部分的に基づいて、前記保存された初期状態又は終了状態の選択された一方から前記プロセスの実行を再開することを更に含む、請求項1に記載のコンピュータにより実施される方法。

【請求項20】

前記プロセスを初期化し前記プロセスの前記初期状態に関する情報を保存した実質的に直後の第1の実行フェーズ中に前記アボートコマンドメッセージを受信することと、 30

前記プロセスをシャットダウンして再開する必要なく、前記保存された初期状態から前記プロセスの実行を再開することと、

を更に含む、請求項17に記載のコンピュータにより実施される方法。

【請求項21】

初期化の時にプロセスの初期状態に関する情報を記憶することであって、前記プロセスの実行が少なくとも1つの実行フェーズを実行することを含み、及び前記実行フェーズの前記実行の完了時に前記実行フェーズの終了状態を表す情報を記憶することと、

所定のイベントに応答して前記プロセスの実行をアボートすることと、 40

前記プロセスをシャットダウンする必要なく、前記保存された初期状態及び終了状態の一方から前記プロセスの実行を再開することと、

をコンピューティングシステムに実行させるための命令を含むコンピュータプログラムを記憶するコンピュータ読み取り可能な記憶媒体。

【請求項22】

初期化の時にプロセスの初期状態に関する情報を受信し記憶するように構成され、前記プロセスの実行が少なくとも1つの実行フェーズを実行することを含み、更に、前記実行フェーズの前記実行の完了時に前記実行フェーズの終了状態を表す情報を記憶する、入力デバイス又はポートと、

少なくとも1つのプロセッサであって、 50

所定のイベントに応答して前記プロセスの実行をアボートし、
前記プロセスをシャットダウンする必要なく、前記保存された初期状態及び終了状態
の一方から前記プロセスの実行を再開する、
ように構成された少なくとも1つのプロセッサと、
を含むコンピューティングシステム。

【請求項23】

初期化の時にプロセスの初期状態に関する情報を記憶し、前記プロセスの実行が少なくとも1つの実行フェーズを実行することを含み、更に、前記実行フェーズの前記実行の完了時に前記実行フェーズの終了状態を表す情報を記憶するための手段と、
前記プロセスの実行を制御するための手段であって、前記制御が、

10

所定のイベントに応答して前記プロセスの実行をアボートすることと、

前記プロセスをシャットダウンする必要なく、前記保存された初期状態及び終了状態
の一方から前記プロセスの実行を再開することと、
を含む、手段と、
を含むコンピューティングシステム。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本出願は、2011年2月18日に出願された「Restarting Processes」と題する米国
特許出願第13/030,998号に対する優先権を主張する。その全体的な内容は参照
により本願にも含まれるものとする。

20

【0002】

本記載はプロセスの再開に関する。

【背景技術】

【0003】

シングルプロセッサコンピュータによって提供される計算速度は、過去数十年間で飛躍的に進歩している。しかしながら、かかるプロセッサにより実行される多くのアプリケーションが必要とする計算容量は、最速のシングルプロセッサコンピュータをも超える場合がある。例えば、航空予約システムのようなトランザクショナルシステムにおいて、多数のユーザがコンピュータリソースに同時にアクセスすることがある。こういったユーザは通常、応答時間が短いことを期待している。シングルプロセッサコンピュータは、かかる要求に追いつくことができない恐れがある。このようなアプリケーションを処理して性能を向上させるために、並列処理システム等の多種多様なアーキテクチャが開発されている。一般に、並列処理システムによって用いられる多数のプロセッサは、単一の場所に配置するか又は遠隔分散させることができる。場合によっては、本質的に連続的かつリアルタイムに近い処理を含む大量のデータを処理するアプリケーションでは、処理能力のため、こういった並列処理システムに頼るようになっている。かかる処理能力は、システム障害に対してロバスト(robust)でレジスタンス(resistance)であること、すなわちフォルトトレラントであることが期待される。

30

【発明の概要】

【発明が解決しようとする課題】

【0004】

これらの能力は、大規模なインターネットベースのデータ処理からプライベートなネットワーク及び通信システム(例えば企業内「イントラネット」等)までに及ぶあらゆる種類及びサイズのコンピュータネットワークにとって有用である。

【課題を解決するための手段】

【0005】

一態様において、概して、コンピュータにより実施される方法は、初期化の時にプロセスの初期状態に関する情報を記憶することであって、プロセスの実行が少なくとも1つの

40

50

実行フェーズを実行することを含み、及び実行フェーズの実行の完了時に実行フェーズの終了状態を表す情報を記憶することと、所定のイベントに応答してプロセスの実行をアボートすることと、プロセスをシャットダウンする必要なく、保存された初期状態及び終了状態の一方からプロセスの実行を再開することと、を含む。

【0006】

複数の態様は以下の1つ以上を含むことができる。

【0007】

所定のイベントは、外部デバイスに対する接続の切断である場合がある。所定のイベントは、外部デバイスによるエラーである。外部デバイスに対する接続が復元された場合に、プロセスの実行を再開することができる。外部デバイスによるエラーがクリアされた場合に、プロセスの実行を再開することができる。所定のイベントが発生した実行フェーズの前に記憶された終了状態から、プロセスの実行を再開することができる。プロセスの起動の実質的に直後の実行フェーズ中に所定のイベントが発生した場合、初期状態からプロセスの実行を再開することができる。実行フェーズの実行は、受信したデータストリームに1つ以上の処理アクションを実行して実行フェーズに対応した出力データを生成することを含むことができる。

10

【0008】

この方法は、1つ以上の実行フェーズに対応した出力データを記憶することと、プロセスの実行が再開された場合に出力データを再現することと、を更に含むことができる。プロセスは、プロセスステージの一部であり、データ経路を介してプロセスステージにおける第2の異なるプロセスと通信状態にある場合がある。この方法は、プロセスステージにおける各プロセスを介してチェックポイントコマンドメッセージを伝達することと、各プロセスにおいて、チェックポイントコマンドメッセージの受信時に初期状態又は終了状態に関する新しい情報を保存することであって、保存することが、プロセスの動作を中断すること及び新しい情報を記憶領域に保存することを含む、ことと、を更に含むことができる。コンピュータにより実施される方法は、古い保存された初期状態又は終了状態を新しい初期状態又は終了状態で上書きすることを更に含むことができる。

20

【0009】

各プロセスは、プロセスのためのデータを受信しキューに入れるためのデータキューと通信状態にある場合がある。この方法は、トリガイベントの検出に応答してチェックポイントコマンドメッセージを生成することを更に含むことができる。トリガイベントは、ネットワークイベントに関する情報を含むことができる。この方法は、チェックポイントコマンドメッセージを定期的に生成することを更に含むことができる。

30

【0010】

この方法は、処理している受信データレコード内の又はこのデータレコードから導出される選択されたデータ値の発生に応答してチェックポイントコマンドメッセージを生成することを更に含むことができる。この方法は、所定のイベントに応答してアボートコマンドメッセージを発生することと、プロセスステージにおける各プロセスを介してアボートコマンドメッセージを伝達することと、各プロセスにおけるアボートコマンドメッセージの受信時に、プロセスの実行をアボートし、アボートコマンドメッセージを次のプロセスに伝達することと、を更に含むことができる。この方法は、処理再開メッセージに含まれた情報に部分的に基づいて、保存された初期状態又は終了状態の選択された一方からプロセスの実行を再開することを更に含むことができる。この方法は、プロセスを初期化しプロセスの初期状態に関する情報を保存した実質的に直後の第1の実行フェーズ中にアボートコマンドメッセージを受信することと、プロセスをシャットダウンして再開する必要なく、保存された初期状態からプロセスの実行を再開することと、を更に含むことができる。

40

【0011】

別の態様において、概して、コンピュータプログラムを記憶するコンピュータ読み取り可能記憶媒体は、初期化の時にプロセスの初期状態に関する情報を記憶することであって

50

、プロセスの実行が少なくとも 1 つの実行フェーズを実行することを含み、及び実行フェーズの実行の完了時に実行フェーズの終了状態を表す情報を記憶することと、所定のイベントに応答してプロセスの実行をアボートすることと、プロセスをシャットダウンする必要なく、保存された初期状態及び終了状態の一方からプロセスの実行を再開することと、をコンピューティングシステムに実行させるための命令を含む。

【 0 0 1 2 】

別の態様において、概して、コンピューティングシステムは、初期化の時にプロセスの初期状態に関する情報を受信し記憶するように構成され、プロセスの実行が少なくとも 1 つの実行フェーズを実行することを含み、更に、実行フェーズの実行の完了時に実行フェーズの終了状態を表す情報を記憶する、入力デバイス又はポートと、少なくとも 1 つのプロセッサであって、所定のイベントに応答してプロセスの実行をアボートし、プロセスをシャットダウンする必要なく、保存された初期状態及び終了状態の一方からプロセスの実行を再開する、ように構成された少なくとも 1 つのプロセッサと、を含む。

10

【 0 0 1 3 】

別の態様において、概して、コンピューティングシステムは、初期化の時にプロセスの初期状態に関する情報を記憶し、プロセスの実行が少なくとも 1 つの実行フェーズを実行することを含み、更に、実行フェーズの実行の完了時に実行フェーズの終了状態を表す情報を記憶するための手段と、プロセスの実行を制御するための手段であって、制御が、所定のイベントに応答してプロセスの実行をアボートすることと、プロセスをシャットダウンする必要なく、保存された初期状態及び終了状態の一方からプロセスの実行を再開することと、を含む、手段と、を含む。

20

【 0 0 1 4 】

複数の態様は以下の利点の 1 つ以上を含むことができる。

【 0 0 1 5 】

マルチプロセス処理システムにおけるプロセスは、別個の実行フェーズにおいて実行することができる。システム障害のイベントでは、処理システムを終結させて最後に完了したチェックポイントから再開することは、必要以上の量の処理時間及びリソースを消費する可能性がある。例外条件に応答して処理システムがそのアクティビティを終結させた後、処理システムは、かかるシステムの経験者である情報技術の専門家によって手作業で再初期化しなければならない場合がある。この結果、著しいシステムダウンタイムとなることがある。ある例では、システム障害を検出して専門家に通知するために別個のプロセスを設計する必要がある場合がある。このため、効率を向上させると共に処理リソースの消費を軽減するため、処理システム内のプロセスは、プロセスに対する接続障害の復元時に、システム全体を再開するのではなく、最後に記録したチェックポイントから実行することができる。1 つの実施においては、システム全体を終結させて再開するのではなく、接続障害が復元されるまで処理を中断することをシステム内の個々のプロセスに通知することができる。

30

【 0 0 1 6 】

本発明の他の特性及び利点は、以下の記載から及び特許請求の範囲から明らかとなろう。

40

【 図面の簡単な説明 】

【 0 0 1 7 】

【 図 1 】マルチプロセスデータ処理システムのブロック図である。

【 図 2 】例示的なマルチプロセスデータ処理システムを示す。

【 図 3 】例示的なマルチプロセスデータ処理システムを示す。

【 図 4 】例示的なチェックポイントティングプロセスを示すフローチャートである。

【 図 5 】例示的な回復機構のフローチャートである。

【 図 6 】例示的な回復機構のフローチャートである。

【 発明を実施するための形態 】

【 0 0 1 8 】

50

図1を参照すると、データ処理システム100は、データを処理するために合理的に配置された多数のプロセスを提供する。例示的なシステム100内で、データは、データソース102（例えばウェブサーバとして機能するサーバ104上で実行されているアプリケーション）から受信される。そして、データは、コンピュータシステム108上で実行されているか又は分散して（例えば2つ以上のネットワーク化コンピュータ端末によって）実行されているマルチプロセスデータ処理モジュール106に伝達される。データ処理モジュール106は、システム100のデータ処理の側面を監視し、制御し、実行する。かかる処理を提供するため、データ処理モジュール106は、1つ以上のプロセス116、118による処理対象のデータを記憶することができる1つ以上のキュー110、112、114を含む。この例では、図示のように、データソース102から受信されたデータは、初期データキュー110に記憶され、定期的に初期プロセス116に与えられる。プロセス116は、データを処理（例えば変換、フィルタリング、内容確認等）し、処理したデータを1つ以上の下流データキュー112、112'に提供する。次のプロセス118、118'は、キュー112、112'からのデータが与えられ、他の（又は同様の）処理を実行することができ、この後でその結果を他の下流データキュー114、114'に送出する。例示するデータ処理モジュール106のキュー及びプロセスのレイアウトは、利用可能な多くの考えられる処理スキームの1つである。例えば、データ処理モジュール106は、図示するプロセスの上流、下流、又はこのプロセスとは無関係に配置することができる追加プロセス（例えば並列又は直列の実行のため）を含むことも可能である。ある例では、最後のキューセット（例えばキュー114及び114'）からのデータは、リレーショナルデータベース管理システム（RDBMS）等の宛先アプリケーション120（又は多数のアプリケーション）に出力することができる。

10

20

30

40

50

【0019】

データ処理モジュール106に含まれるプロセスは、外部デバイス及び／又は他の処理システム（例えばコンピュータシステム122）と通信状態にすることができる。例えば、プロセスは、他のシステムからこのプロセスにメッセージを提供するJMS（Java（登録商標）Message Service）キューと通信状態にすることができる。ある例では、データ処理モジュール106内のプロセスは、（例えば外部システム122に配置された）1つ以上のデータベースと通信状態にすることができる。例えば、モジュール106は、1つ以上の現金自動預払機（ATM）において対応する顧客のセッションから受信された情報に基づいて、銀行データベース内の顧客の金融口座に対する更新を実行することができる。

【0020】

一例として、図2は、処理システム200を示し、これは、遠隔で実行される処理モジュール202（ATM204により実行される）を有する。モジュール202は、中央位置で処理するデータを提供するために用いられている。図示する例では、例えばATM読み出しプロセス206のような初期プロセスが、ATMから（例えばトランザクションに関連付けられた）顧客の口座データを受信し、口座の詳細を認証するためにこのデータを口座確認プロセス208に渡すことができる。この例では、口座確認プロセス208は、顧客が個人識別番号（PIN）データベース210に対して入力したPINを検証することができる。一旦、顧客の本人確認が認証されると、更にデータレコードを下流の残高チェックプロセス212に伝達することができる。プロセス212は、例えば識別した顧客口座の残高をチェックするために第2の異なるデータベース214と通信を行うことができる。更に別のトランザクションが完了した後、追加データを下流の残高更新プロセス216に送信することができる。プロセス216は、例えば顧客口座に関連付けられた残高情報を更新するために第3のデータベース218と通信を行うことができる。回答生成プロセス220では、トランザクションの出力概要を用意することができ、これを（例えば顧客に対してATM204上で表示するために）出力表示プロセス222に提供することができる。システムレベルのモニタリング（例えばシステム品質保証）又は他の用途では、データベース210、214、及び218は、マスターデータサーバ224と通信状態

にすることができる。ある実施において、データベースは、例えばスタンダロンのコンピュータシステム 226 によって実行されてもよい。

【0021】

かかるマルチプロセス処理システムにおけるプロセスは、別個の実行フェーズにおいて実行することができる。プロセスの実行は、別個の実行フェーズにおけるプロセス内での1つ以上のタスクの実行を含むことができる。このようなフェーズに実行をセグメント化することによって、別個の実行フェーズを、例えばデータ処理における多数の論理エンドポイント又はブレークポイントによって終結させることができる。各実行フェーズは、その実行フェーズの目的を達成するために1つ以上の処理アクションを有する場合がある。一例として、口座確認プロセス 208 は、1つ以上的方法で別個の実行フェーズで実行することができる。例えば第1の実行フェーズとして、口座確認プロセス 208 は最初に顧客から個人識別番号 (PIN) 情報を受信することができる。顧客から PIN 情報を受信する際の様々な処理アクションは、例えば、ATM ディスプレイ上にプロンプトを表示すること、及び PIN 情報のデータ入力を検証するためのルーチンを実行することを含むことができる。次の実行フェーズでは、プロセス 208 は、データベース 210 に対する接続を確立し、PIN 情報をキーとして用いて顧客のレコードを識別することができる。一旦、プロセス 208 が、顧客のレコードとのトランザクションを完了したら、データベース 210 に対する接続を終結させることができる。最後の実行フェーズでは、プロセス 208 は、前述のトランザクションに基づく結果を発生することができる。このように、これらの実行フェーズの各々は、口座確認プロセス 208 を一時的に中断及び/又は再開することができる別個の論理エンドポイント (又は処理ブレークポイント) を含む。

10

20

30

40

【0022】

ある状況において、正常なシステム動作の過程に影響を及ぼす傾向がある1つ以上のイベントが発生することがある。かかるイベントは、処理システムに含まれるハードウェアモジュール又はソフトウェアモジュールのいずれかにより生じる例外又はエラーであり得る。例えば、ハードウェアの例外又はエラーは、1つ以上のハードウェアユニットからのリセット、割込、又は他の信号を含む場合がある。また、ゼロ除算、オーバーフロー、命令復号エラー、未定義命令等の数値的エラーについて、演算論理によって例外が発生する場合がある。1つ以上のそのようなイベントの発生に基づいて、補正アクション (例えば保守、二次システムへの切り換え等) を行うために、データベースの1つ以上の動作を一時的に停止させる必要がある場合がある。

【0023】

動作の停止及び補正アクションを必要とする場合がある他のイベントには、データベース 210 ~ 224 の1つ以上の障害の検出が含まれ得る。かかる障害は、様々な理由で発生する可能性がある。例えば、メモリ割り当てにエラーがあるか、又はメモリ空間への書き込みに競合がある場合に障害が起こり得る。また、プロセスが残高ゼロの口座から資金を引き出そうとした場合等、根本的なデータ動作のエラーでも障害が起こり得る。一時的な障害があるイベントに加えて、オペレータの介入によって誘発されるイベントによっても障害が起こり得る。ある実施では、オペレータは、障害を引き起こした状況を補正することができ、又は、システムがやがてその状況を補正する場合がある。イベントの例は、ネットワークに接続された1つ以上のデバイスの障害、保守のための1つ以上のデバイス又はソフトウェアサービスのシャットダウン、デバイス又はソフトウェアサービスの障害及び切り換え、記憶空間等のリソースの使い果たし、処理ユニットの過負荷、1つ以上のソフトウェアサービスのタイムアウトを含み得るが、これらに限られるものではない。

【0024】

かかるイベントを検出し、これに対処するため、データ処理システムは、チェックポイントティング技法と称されることが多い1つ以上の技法を用いて、障害イベントにおいて又はシステムが保守又は切り換えのためにオフラインとなった場合においてシステムダウンタイムを最小限に抑えることを保証することができる。チェックポイントティング技法は、概して、プロセスの現在の状態の詳細をチェックポイントレコードとして記憶することを

50

含み、プロセスが、後にその状態から再開するために、記憶された情報を用いることができる。例えば、口座確認プロセス 208 は、各実行フェーズの完了時に（次の実行フェーズ又は他の処理の実行を開始する前に）チェックポイントレコードにその現在の状態を保存することができる。

【0025】

チェックポイントレコードは、プロセス値、処理が成功したレコードに関する情報、及びプロセスの現在の実行フェーズに関連した他の詳細等、様々なタイプの情報を含むことができる。例えばチェックポイントレコードは、データが処理されているデータキュ（例えば図 1 のデータキュ 112）内の現在の位置に関する情報を含むことができる。このため、動作の停止後、このキュ位置から処理を再開することができる。これによって、システム障害からの回復後、プロセスは、初期状態から再開するのではなく、記憶した中間チェックポイントから再開することができる。

10

【0026】

一例として、PINデータベース 210 に障害が発生した場合、口座確認プロセス 208 は例外を生じて処理システム全体を終結させることができる。再開時、処理システム内のプロセス（又はプロセスの一部）は、最後のチェックポイント状態から処理を継続することができる。この例では、顧客が自身の PIN 情報を提供した後の時点で、障害及び再開が発生するので、PIN 情報はプロセスに復元され、顧客から PIN 情報を再収集する必要はない。従って、顧客に PIN 情報を提供するように再び要求する必要をなくすことができる。

20

【0027】

システム障害のイベントでは、処理システムを終結させ、最後に完了したチェックポイントから処理システムを再開することは、必要以上の量の処理時間及びリソースを消費する可能性がある。例外条件に応答して処理システムがそのアクティビティを終結させた後、処理システムは、かかるシステムの経験者である情報技術の専門家によって手作業で再初期化しなければならない場合がある。この結果、著しいシステムダウンタイムとなることがある。ある例では、システム障害を検出して専門家に通知するために、別個のプロセスを設計することを必要とすることがある。チェックポイントイングシステムの例は、「Continuous Flow Checkpointing Data Processing」と題する米国特許第 6,584,581 号、「Overpartitioning system and method for increasing checkpoints in component-based parallel applications」と題する米国特許第 5,819,021 号、及び「Methods and Systems for Reconstructing the State of a Computation」と題する米国特許第 5,712,971 号に記載されている。これらの各々の内容はその全体が本願にも含まれるものとする。

30

【0028】

効率を向上させると共に処理リソースの消費を軽減するため、処理システム内のプロセスは、プロセスに対する接続障害の復元時に、システム全体を再開するのではなく、最後に記録したチェックポイントから実行する。1つの実施においては、システム全体を終結させて再開するのではなく、接続障害が復元されるまで処理を中断することをシステム内の個々のプロセスに通知することができる。

40

【0029】

図 3 は、マルチプロセスシステム 300 のブロック図を示す。システム 300 は、データソースプロセス 302、プロセス 304a～n、データシンクプロセス 306、及び他のプロセス（プロセス 302、304a～n）の各々と通信状態にあるフォルトトレランスマネージャ 308 を含む。ある実施において、フォルトトレランスマネージャ 308 は、マルチプロセスシステム 300 内の別のプロセスとして実行することができる。ある状況では、フォルトトレランスマネージャ 308 は、別個のコンピュータシステム（図示せず）上で実行しているか又はチェックポイントプロセッサ等の専用プロセッサで実行されているアプリケーションであってもよい。チェックポイントプロセッサは、例えば「Continuous Flow Checkpointing Data Processing」と題する米国特許第 6,584,581

50

号に記載されている。その内容は全体が本願にも含まれるものとする。

【0030】

1つ以上の技法を実施して、プロセス302～306とフォルトトレランスマネージャ308との間の通信を確立することができる。例えば、プロセス302～306で発生し得る例外条件に関する情報を伝達するために、個々の例外チャネル310a～nを用いることができる。チャネル310a～nは、有線、無線、又は有線及び無線を組み合わせたネットワークシステムの一部とすることができます。チャネル310a～nは、プロセス302～306により用いられて、プロセス302～306に関するエラー情報をフォルトトレランスマネージャ308に伝達することができる。例えば、プロセス304aと通信状態にある外部デバイスに障害が発生した場合、プロセス304aは、直ちにエラーフラグを立てて、このエラーを、例外チャネル310bを介してフォルトトレランスマネージャ308に伝達することができる。10

【0031】

例外チャネル310a～nに加えて、フォルトトレランスマネージャ308は、対応する通信チャネル312a～eを介してプロセス302～306にコマンドメッセージ（例えばチェックポイントコマンドメッセージ）を送信することができる。通信チャネル312a～eは、フォルトトレランスマネージャ308からのコマンドメッセージをプロセス302～306の各々に順次送信するように構成されている。例えば、フォルトトレランスマネージャ308からのメッセージは、最初にデータソースプロセス302に伝達され、次いでチャネル312b～dを介して各々のプロセス304a～n及びデータシンクプロセス306にシリアルに送ることができる。データシンクプロセス306は、チャネル312eを用いてコマンドメッセージをフォルトトレランスマネージャ308に伝達することができる。20

【0032】

各々のプロセス302、304a～n、306に関連付けられた各チェックポイントデータを記憶するため、各プロセスに記憶領域（例えばメモリ）を割り当てることができる。各プロセスは、別個の実行フェーズの終了時に、その現在の動作を定期的に中断し、そのチェックポイントデータを関連付けられた記憶領域に記憶する。例えば、データソースプロセス302は、処理における別個の実行フェーズの終了時に（受信データのストリームにおける論理ブレークポイント等）、その現在の動作を定期的に中断することができ、チェックポイント情報を記憶領域312に記憶する。このように、プロセス302、304a～n、306の各々が実行されると、対応する記憶領域312、314a～n、及び316は、定期的にチェックポイントデータを保存する。記憶領域312～316は、例えば磁気媒体のような不揮発性記憶等、ハードディスクドライブ等の様々なタイプの記憶技法によって実施することができる。ある例では、単一の記憶領域を2つ以上のプロセスによって共有することも可能である。チェックポイントデータは、現在の状態に関する情報及び／又はプロセス302～306に関連付けられたデータを含んで、後の時点でそれらの状態の再構築を可能とすることができる。30

【0033】

ある例では、データ記憶領域312～316は、プロセスに関連付けられたデータキュー（例えば図1のキュー110、112、及び114）も記憶することができる。例えば記憶領域316は、プロセス304nから処理済みデータを受信するためのキューを含むことができ、ここからデータを出力又は公表（例えば印刷又は表示）することができる。40

【0034】

フォルトトレランスマネージャ308は、チェックポイントコマンドメッセージを生成し、通信チャネル312a～eを介して各プロセス302～306に順次送ることによって、チェックポイントティング動作を管理する。チェックポイントコマンドメッセージは、各プロセス302～306を通るので、プロセスは、メッセージを受信するとその現在の状態のチェックポイントを行うことができる。このように、チェックポイントコマンドメッセージは、データソースプロセス302に伝わり、次いで各プロセス304a～n及50

びデータシンクプロセス 306 を順次通り、その後でフォルトトレンラスマネージャ 308 に戻される。このチェックポイントティング動作は、一定間隔で自動的に開始することができる。例えばフォルトトレラスマネージャ 308 は、例えば 5 分ごと等の所定の定期的なペースでチェックポイントコマンドメッセージを送信することができる。定期的なペースは、デフォルト値に設定することができ、又はユーザにより調節することも可能である。

【0035】

システムに含まれるプロセスについてチェックポイントを記憶する動作を開始するために、1つ以上の技法を実施することができる。例えば、1つ以上の外部トリガが、チェックポイント情報を記憶するための動作を開始することができる。一例では、ネットワークメッセージが、フォルトトレラスマネージャ 308 に、今にも起こりそうなネットワークシャットダウンを通知し、これによってチェックポイントティング動作をトリガすることができる。ある実施では、チェックポイントティング動作は、処理しているデータレコード内の又はそれから導出される値に応答してトリガすることができる。例えば、処理済みデータレコードは、チェックポイントティングを実行することができる論理ポイントと考えられるタイムスタンプ又はブレークポイント値を含むことができる。

【0036】

システムによってデータを処理している期間中にチェックポイント情報を記憶することに加えて、データ処理の前に情報を記憶することができる。1つの実施では、マルチプロセスシステム 300 が最初に初期化された時、例えば起動中に、初期チェックポイントティング動作をトリガすることができる。フォルトトレラスマネージャ 308 は、各々のプロセス 302 ~ 306 を経るよう初期チェックポイントコマンドメッセージを送ることができる。図 3 に示す例では、初期チェックポイントメッセージは、最初にデータソースプロセス 302 に伝達される。データソースプロセス 302 は、直ちにチェックポイントを行う。例えば、その初期状態を表すデータを関連付けられたデータ記憶空間 312 に記憶し、初期チェックポイントメッセージを下流の次のプロセス 304a に渡す。この初期チェックポイント状態をチェックポイント状態ゼロと称する。

【0037】

同様に、シリアルに、各々のプロセス 304 ~ 306 は、その初期状態及び関連付けられたデータ値を適切な記憶領域にチェックポイント状態ゼロとして同様に記憶することができる。例として、初期状態及び関連付けられたデータ値は、グローバル変数の初期値、参照データ情報、及びカウンタの初期値を含む監査変数 (auditing variables) を含むことができる。

【0038】

各々のプロセス 302 ~ 306 が、その初期状態を記憶した後、初期チェックポイントコマンドメッセージは、チャネル 312e を介してフォルトトレンラスマネージャ 308 に戻される。プロセス 302、304a ~ n、306 を通って一周した後にメッセージをフォルトトレラスマネージャ 308 に戻すことに基づいて、フォルトトレラスマネージャは、プロセス 302 ~ 306 がチェックポイント状態ゼロを完了したことを通知される。ある実施では、下流プロセスが現在の状態を保存している間、ソース及び他の上流プロセスは、全てのプロセスが状態を保存するのを待つことなく、データの受信及び他の機能の実行を続けることができる。

【0039】

同様に、プロセス 302 ~ 306 の各別個の実行フェーズについて、追加のチェックポイントティングを実行可能である。このため、初期チェックポイント情報を表すデータを記憶することに加えて、フォルトトレンラスマネージャ 308 は、例えば以降のチェックポイントサイクル (例えばチェックポイント状態 1、2、3、... n) に関連付けられた情報を表す追加情報の記憶を開始することができる。以降のチェックポイント情報の記憶を開始するため、プロセス 302、304a ~ n、306 に更に別のチェックポイントコマンドメッセージを伝搬する等の技法を利用することができる。チェックポイントコマ

10

20

30

40

50

ンドメッセージを受信すると、プロセス 304a は、進行中のタスクを完了するか、又は未完了のタスクを中断することができる。ある例では、プロセス 304a は、データ記憶 314 に記憶された以前に生成したチェックポイントレコードを削除し、記憶空間を取り戻すことができる。次いでプロセス 304 は、その現在の状態及び関連付けられたデータのための新しいチェックポイントレコードを生成することができる。ある状況では、チェックポイントレコードは、メモリに永続的に記憶されるか、又は新しいチェックポイントレコードによって上書きされる。チェックポイントレコードに記憶される情報の更に別の例が米国特許第 6,584,581 号に与えられている。その内容は全体が本願にも含まれるものとする。

【0040】

10

ある実施において、プロセス 304a は、(コンピュータシステム 320 上で実行される) 外部データベース 318 と更に通信状態にすることができる。時に、データベース 318 に対する接続に障害が発生する場合があり、又はデータベース 318 が保守のためにオフラインとなる場合がある。この障害は、データベース 318 を実行しているコンピュータシステム 320 のハードウェア障害であり得る。かかる状況では、プロセス 304a は、例外チャネル 310b を介してエラーフラグを立てて、接続が失われたことをフォルトトレランスマネージャ 308 に通知することができる。

【0041】

20

エラーの通知を受信すると、フォルトトレランスマネージャ 308 は、アボート (abort) コマンドメッセージを生成して、このコマンドを、プロセス 302 ~ 306 を介して伝搬させることができる。アボートコマンドメッセージは、典型的に、最初にチャネル 312a によってデータソースプロセス 302 に伝達され、次いでチャネル 312b ~ d によって各々のプロセス 302 ~ 306 に送られ、最後にチャネル 312e によってフォルトトレランスマネージャ 308 に戻される。各々のプロセス 302 ~ 306 は、アボートコマンドメッセージを受信すると、(もしあれば) 比較的小さい遅延でその現在のアクティビティをアボートし、最後のチェックポイント状態の後に処理されたかもしれない未完了のタスク又はレコードを流す (flush) / 処分する。プロセスは、アクティビティをアボートした後、アボートコマンドメッセージを次の下流プロセスに渡すことができる。このように、アボートコマンドメッセージは、シンクプロセス 306 まで伝搬し、その後でフォルトトレランスマネージャ 308 に戻される。フォルトトレランスマネージャ 308 は、シンクプロセス 306 からアボートコマンドメッセージを受信するまで待機し、これによって、プロセス 302 ~ 306 の全てが現在の処理タスクをアボートした(例えば休止状態にある)ことを確実とする。

30

【0042】

40

コンピュータシステム 320 におけるハードウェア障害のために、データベース 318 に障害が発生した状況では、プロセス 302 ~ 306 は、処理をアボートするように指示される。ある実施では、システムがその処理を完全にアボートした後、プロセス 302 は、指定可能な時間量(障害を補正するために要する平均時間量を反映するはずの時間)だけ待機し、最後に保存されたチェックポイント状態から再び処理を開始する。ある実施では、プロセス 304a は、定期的にデータベース 318 をポーリングして、その状態をチェックする(すなわちデータベース 318 が動作可能であるか否かをチェックする)ことができる。ある例では、コンピュータシステム 320 は、データベース 318 が動作可能状態に復元された場合にプロセス 304a に自動的に通知するように構成することができる。処理システム 300 は、データベース 318 との接続が復元されると、最後に保存されたチェックポイント状態から再び処理を開始することができる。

【0043】

50

この点に関して、プロセス 304a は、フォルトトレランスマネージャ 308 に接続が復元されたことを通知する。フォルトトレランスマネージャ 308 は、各々のプロセス 302 ~ 306 について最後に正常に完了したチェックポイント状態を判定し、各々のプロセス 302 ~ 306 に処理再開メッセージを送信する。他のコマンドメッセージと同

様に、処理再開メッセージは、通信チャネル 312a～e を介して各々のプロセス 302～306 に順次伝搬する。処理再開メッセージは、プロセス 302～306 が処理を再開するチェックポイント状態を指定する。

【0044】

ある例では、フルトトレランスマネージャ 308 は、前のチェックポイント動作が現在実行されている間に追加のチェックポイント動作を開始することができる。例えば、プロセス 304n が任意のチェックポイント状態（例えばチェックポイントコマンドメッセージ N に対応するチェックポイント N）を処理している間に、フルトトレランスマネージャ 308 は、次のチェックポイントコマンドメッセージ N+1 を生成してソースプロセス 302 に送信することによって次のチェックポイント状態（例えばチェックポイント状態 N+1）を開始することができる。これによって、チェックポイントコマンドメッセージ N がまだプロセス 302～306 を伝わっている時に、新しい次のチェックポイントコマンドメッセージ N+1 を生成してプロセス 302～306 に送信することができる。このように、フルトトレランスマネージャ 308 によって、前のチェックポイントティング状態が完了するまで待つ必要なく、プロセス状態のチェックポイントティングの頻度を高くすることができる。

【0045】

ある状況では、1つ以上のチェックポイントコマンドメッセージがプロセス 302、304a～n、306 を移動している間にシステム障害が発生し得る。例えば、フルトトレランスマネージャ 308 が、チェックポイントコマンドメッセージ N を発生することでチェックポイント状態 N を開始したという状況について考える。チェックポイントコマンドメッセージ N がプロセス 302～306 により処理されている間、プロセスの1つ（例えばプロセス 304a）と外部システム（例えばデータベース 312）との間の接続に障害が発生する場合がある。この状況が警告されると、フルトトレランスマネージャ 308 は、これに応答して、プロセス 302～306 にアボートコマンドメッセージを送信することができる。アボートコマンドメッセージは、まだチェックポイント状態 N を処理している（例えばチェックポイント N に関連付けられたチェックポイント情報を記憶している）プロセス（例えばプロセス 304n）に到達することができる。アボートコマンドの受信に基づいて、プロセス 304n は、1つ以上のアクションを実行することができる。例えばプロセス 304n は、チェックポイント状態 N を完了させ、全ての更に別の処理をアボートすることができる。別の状況では、プロセス 304n は、前のチェックポイント状態 N-1 以後の現在の状態及び以降の状態に関連付けられた結果を直ちに処分し、更に別の処理をアボートすることができる。この結果、システム 300 が休止となった場合、各々のプロセス 302～306 は、異なるチェックポイント状態にあることがある。例えば、プロセス 304n の上流にある全てのプロセス（例えばデータシンクプロセス 306）は、チェックポイント状態 N を完了しており、一方でプロセス 304n の下流にある全てのプロセス（例えばプロセス 304a 及びデータソースプロセス 302）は、チェックポイント状態 N-1 のみを完了している場合がある。

【0046】

システム 300 が処理を再開する準備ができたら、フルトトレランスマネージャ 308 は、通信チャネル 312a～e を介して各プロセスに1つ以上の処理再開メッセージを送信する。処理再開メッセージは、プロセスに対して、それらが実行するべきである最も早い完全にコミットされた（又は完了した）チェックポイントされた状態（例えばチェックポイント状態 N-1）を示す。ある例では、すでにチェックポイント状態 N を完了したプロセスは、チェックポイント状態 N-1 からチェックポイント状態 N へ、その結果を単に再現することができる。このように、プロセス 302～306 は、以前の作業の繰り返しを回避することができる。ある例では、チェックポイント状態 N-1 からチェックポイント状態 N への結果の再現は、2つのチェックポイント状態間に発生したかもしれない以前の処理アクションの結果を再現することを含む。

【0047】

10

20

30

40

50

例として、実質的に起動直後に、システム障害が発生することがある。かかる状況では、プロセス 302～306 の多くがチェックポイント状態ゼロのみを完了している場合がある。これらのプロセス 302～306 は、対応するチェックポイントレコードに記憶された初期化データ及び起動値に基づいてチェックポイント状態ゼロから処理を再開することができる。

【0048】

図 4 は、マルチプロセスシステム内のプロセス（例えば図 3 のプロセス 302）の例示的な実行を示すフローチャートである。起動時に、プロセスは、その初期状態をチェックポイント状態ゼロとしてデータ記憶部に直ちに記憶する（ステップ 402）。プロセスは次いで別個の実行フェーズ（例えば実行フェーズ 1、2、…、N-1）で実行することができる。各実行フェーズの完了時に、プロセスはその終了状態をデータ記憶部に保存することができる。例えば、終了状態をチェックポイント状態として記憶することができる。例えば第 1 の実行フェーズ後、プロセスは、第 1 の実行フェーズの終了状態をチェックポイント状態 1 として保存することができる（ステップ 404）。同様に、以降の実行フェーズ後、プロセスは、実行フェーズの終了状態をチェックポイント状態 2、…、N-1、及び N として保存することができる（ステップ 406～410）。

【0049】

図 5 は、外部システムに障害が発生したか又は保守のためにオフラインとなった場合に実行される例示的なステップを示すフローチャートである。例えば、外部システムは、処理システム内のプロセスと通信状態にあるデータベース（例えば図 3 のデータベース 312）であってもよい。外部システムが保守のためにオフラインとなるか又はこれに障害が発生すると、例えば障害が発生した外部システムと通信状態にあるプロセスによって、エラーフラグが立てられる（ステップ 502）。エラーフラグに応答して、アボートコマンドメッセージが生成され、プロセスを介して伝搬される（ステップ 504）。プロセスがアボートコマンドメッセージを受信すると、各プロセスの現在のアクティビティがアボートされる（ステップ 506）。また、最後のチェックポイント状態以降に実行されたあらゆるトランザクションをプロセスによって処分することができる。次いで、外部システムに対する接続障害が復元されるまで更に別のアクションはアボートされる（ステップ 508）。接続が復元されると、処理再開メッセージが各プロセスに送信される（ステップ 510）。処理再開メッセージは、プロセスが処理を再開するべきであるチェックポイント状態を示す。このため、各プロセスは、関連付けられた記憶領域からチェックポイント状態に関する関連情報を検索する（ステップ 512）。

【0050】

図 6 は、プロセスの実行中にチェックポイントを記憶し、このチェックポイントから再開する際に実行される例示的なステップを示すフローチャートである。例えば、プロセスの初期化時に、プロセスの初期状態に関する情報が、関連付けられた記憶領域に記憶される（ステップ 602）。次いでプロセスは、別個の実行フェーズで実行される。このため、各実行フェーズの終了時に、プロセスは、実行フェーズの終了状態を表す情報を記憶する（ステップ 604）。所定のイベントが発生すると、例えば外部デバイスに対する接続が失われると（ステップ 608）、プロセスの実行がアボートされる（ステップ 606）。その間、プロセスは、アボートをトリガしたイベントがクリアされた（例えば外部デバイスに対する接続の復元）か否かを調べるためにチェックを行う。この時間中、プロセスは、シャットダウンされず、イベントがクリアされたと見なされるまで処理がアボートされる。プロセスの実行は、最後に保存された初期状態又は終了状態から再開される（ステップ 610）。

【0051】

上述の技法は、コンピュータ上で実行するためのソフトウェアを用いて実施可能である。例えばソフトウェアは、1つ以上のプログラミングされたか又はプログラマブルなコンピュータシステム（これは、分散型、クライアント/サーバ、又はグリッド等の様々なアーキテクチャのものであり得る）で実行する1つ以上のコンピュータプログラムにおいて

10

20

30

40

50

手順を形成し、その各々のコンピュータシステムは、少なくとも 1 つのプロセッサ、少なくとも 1 つのデータ記憶システム（揮発性及び不揮発性メモリ及び／又は記憶要素を含む）、少なくとも 1 つの入力デバイス又はポート、及び少なくとも 1 つの出力デバイス又はポートを含む。ソフトウェアは、例えばデータフローグラフの設計及び構成に関連した他のサービスを提供するもっと大型のプログラムの 1 つ以上のモジュールを形成することができる。グラフのノード及び要素は、コンピュータ読み取り可能媒体に記憶されたデータ構造又はデータレポジトリに記憶されたデータモデルに合致する他の組織化データとして実施することができる。

【 0 0 5 2 】

ソフトウェアは、汎用又は特殊目的のプログラマブルコンピュータによって読み取り可能な C D - R O M 等の記憶媒体上で提供するか、又はソフトウェアが実行されるコンピュータに対するネットワークの通信媒体を介して送出する（伝搬信号に符号化されている）ことができる。機能は全て、特殊目的コンピュータ上で、又はコプロセッサ等の特殊目的ハードウェアを用いて実行可能である。ソフトウェアは分散型で実施して、ソフトウェアによって指定される計算の異なる部分を異なるコンピュータによって実行することができる。そのような各コンピュータプログラムは、好ましくは、汎用又は特殊目的のプログラマブルコンピュータによって読み取り可能な記憶媒体又はデバイス（例えば固体メモリもしくは媒体、又は磁気もしくは光媒体）上に記憶するか又はこれにダウンロードして、この記憶媒体又はデバイスがコンピュータシステムによって読み出されて本明細書に記載した手順を実行する場合にコンピュータを構成し動作させる。また、本発明のシステムは、コンピュータプログラムを用いて構成されるコンピュータ読み取り可能記憶媒体として実施されるものとして考えることができ、そのように構成された記憶媒体によってコンピュータシステムは特定の既定の方法で動作して本明細書に記載された機能を実行する。

【 0 0 5 3 】

本発明の多数の実施形態について記載した。しかしながら、本発明の精神及び範囲から逸脱することなく様々な変形を実施可能であることは理解されよう。例えば、上述のステップのいくつかは順序に依存せず、従って記載したものとは異なる順序で実行可能である。

【 0 0 5 4 】

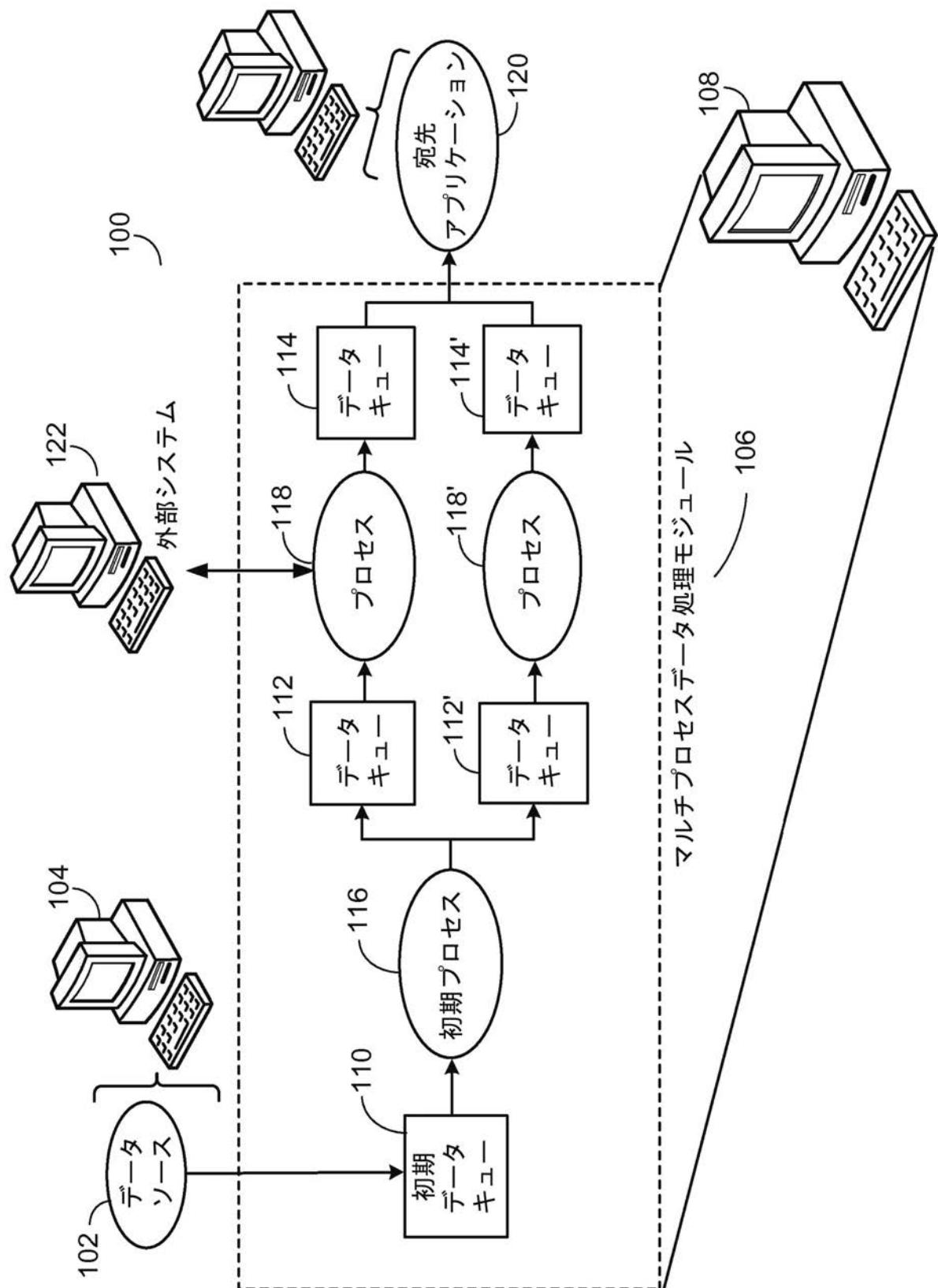
前述の記載は、添付の特許請求の範囲によって規定される本発明の範囲を例示することを意図しており、これを限定することは意図していないことは理解されよう。例えば、上述の多数の機能ステップは、全体的な処理に著しい影響を及ぼすことなく異なる順序で実行可能である。他の実施形態も以下の特許請求の範囲内にある。

10

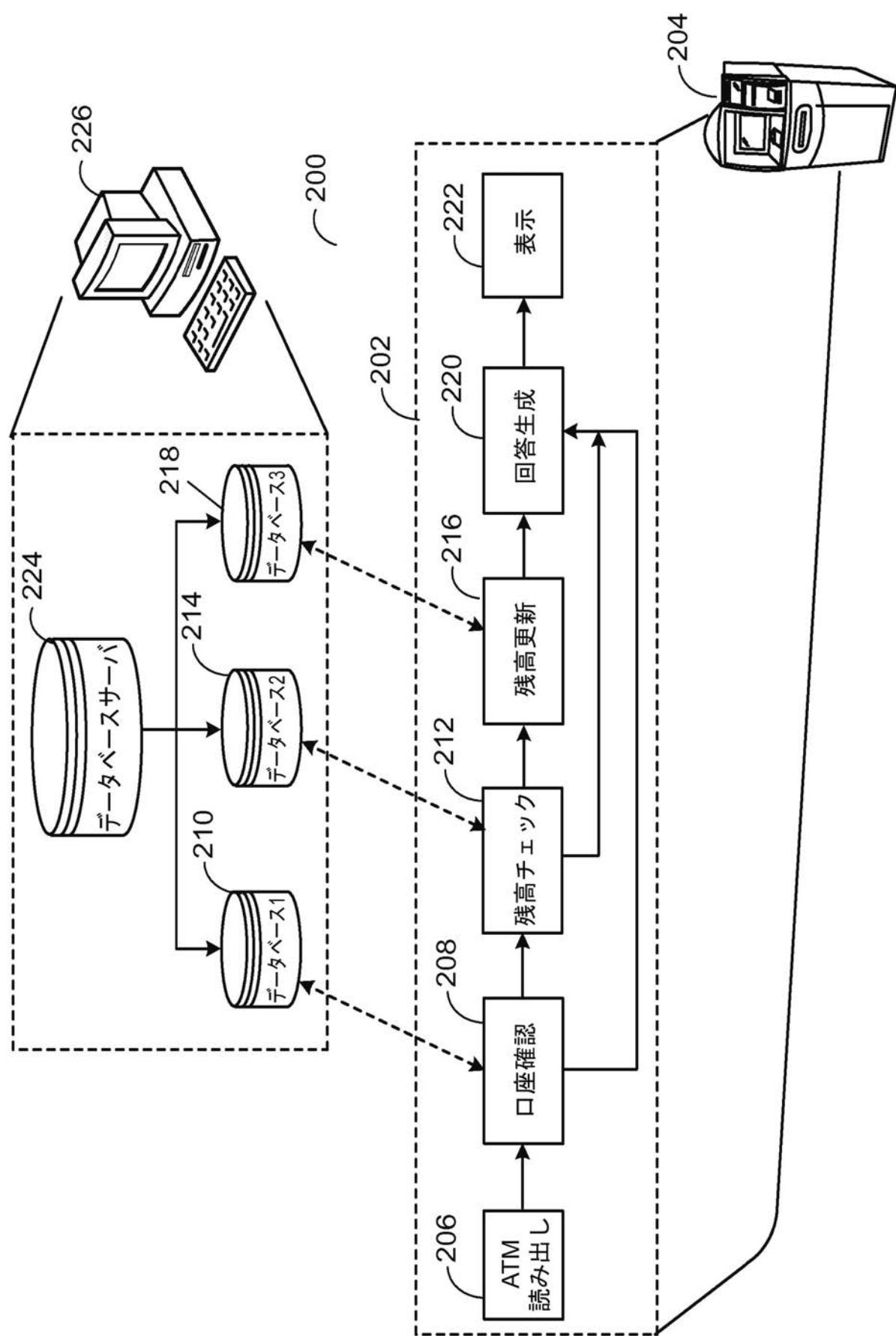
20

30

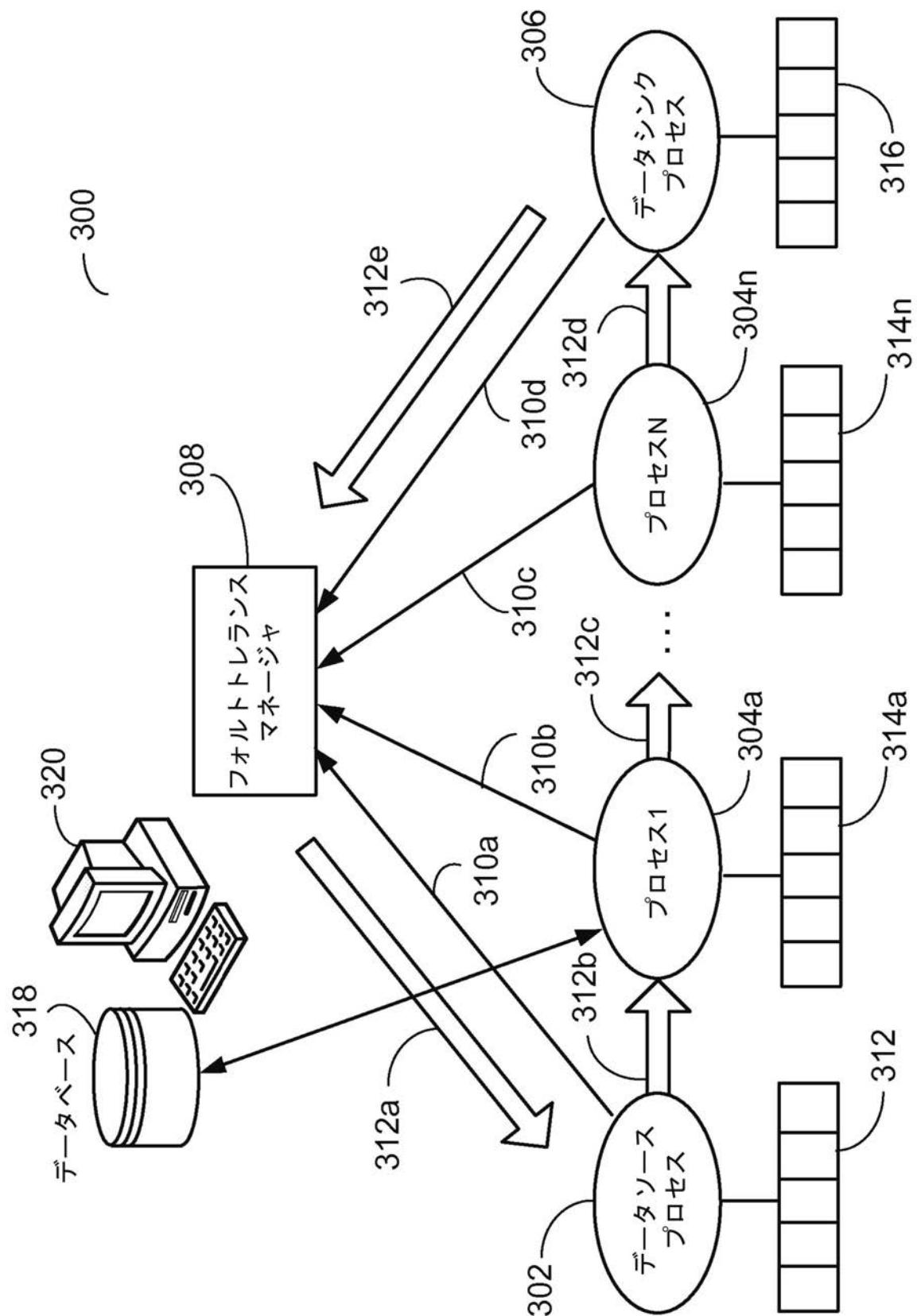
【図1】



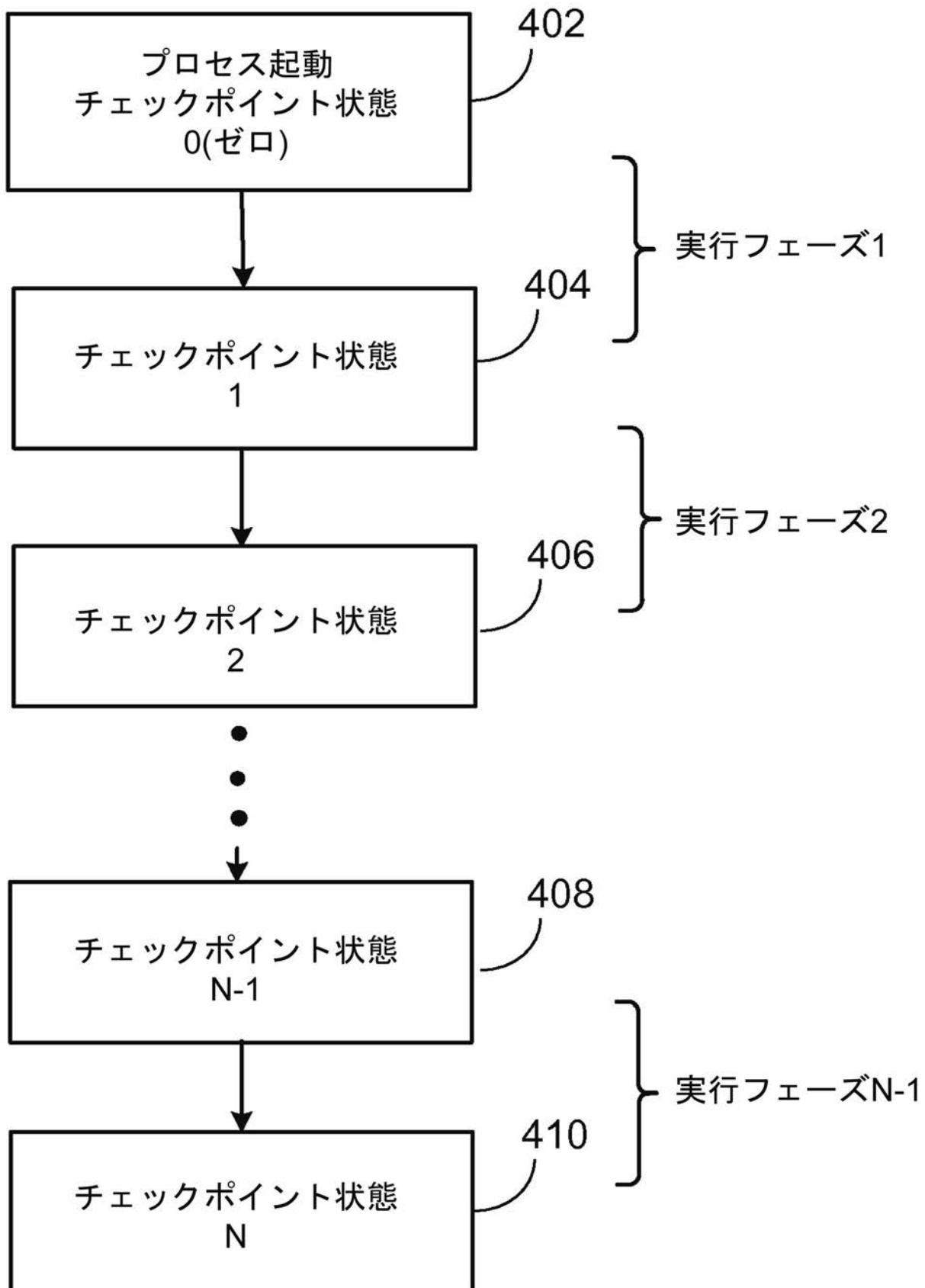
【図2】



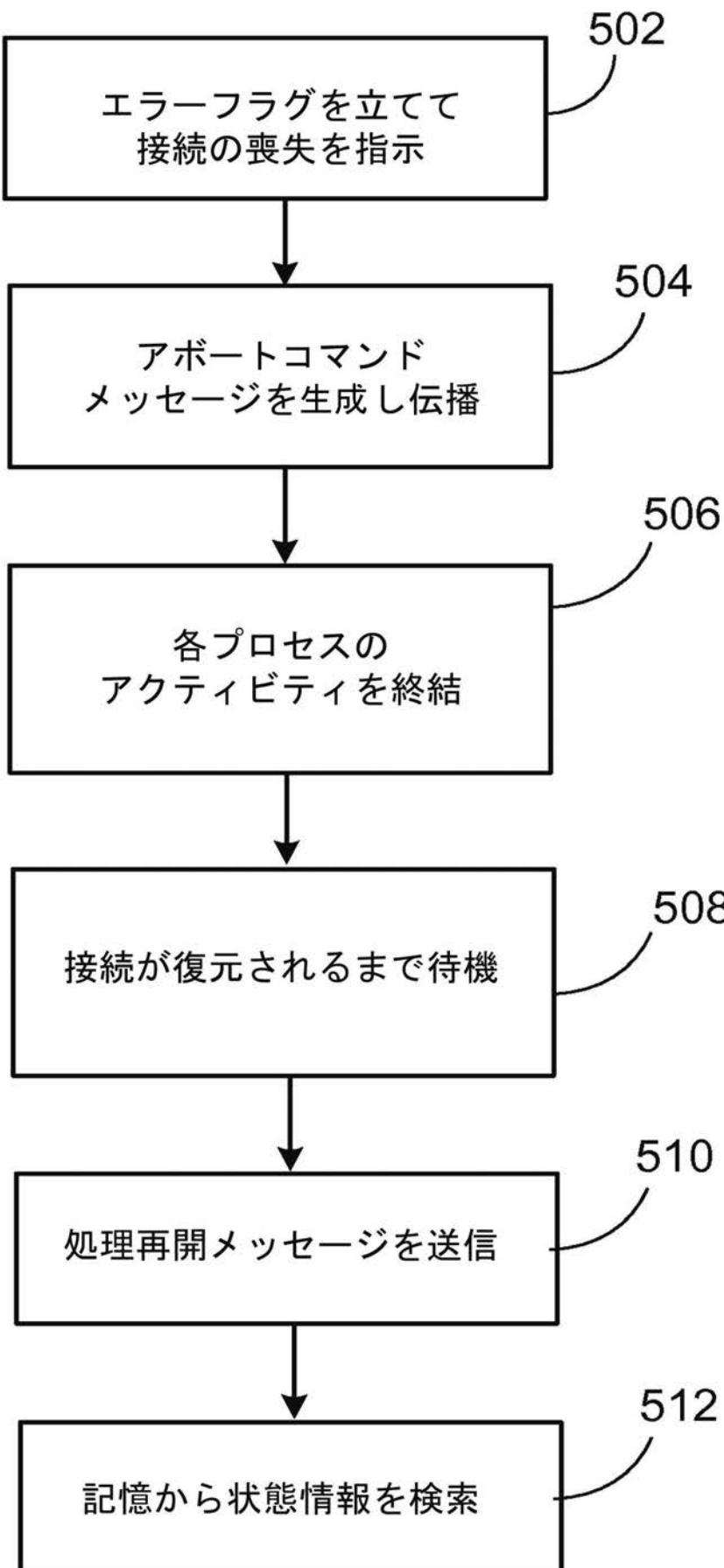
【 図 3 】



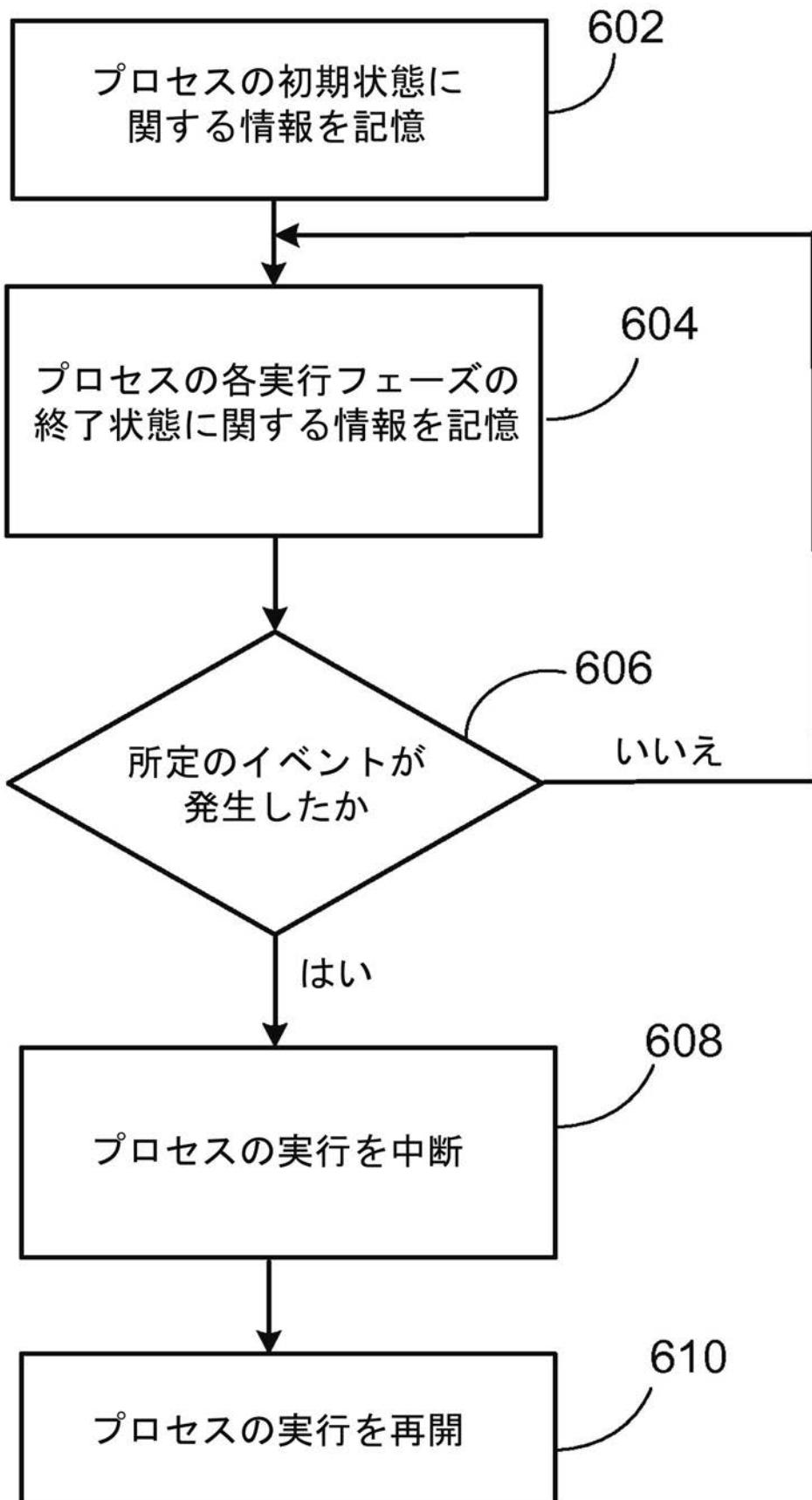
【図4】



【図5】



【図 6】



【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No PCT/US2012/025388						
A. CLASSIFICATION OF SUBJECT MATTER INV. G06F11/14 ADD. G06F9/48								
According to International Patent Classification (IPC) or to both national classification and IPC								
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06F								
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched								
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal								
C. DOCUMENTS CONSIDERED TO BE RELEVANT <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Category^a</th> <th style="width: 80%;">Citation of document, with indication, where appropriate, of the relevant passages</th> <th style="width: 10%;">Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td>X</td> <td> R. BALDONI ET AL: "Characterization of consistent global checkpoints in large-scale distributed systems", PROCEEDINGS OF THE FIFTH IEEE COMPUTER SOCIETY WORKSHOP ON FUTURE TRENDS OF DISTRIBUTED COMPUTING SYSTEMS, 1 January 1995 (1995-01-01), pages 314-323, XP55026754, DOI: 10.1109/FTDCS.1995.525000 ISBN: 978-0-81-867125-8 page 314, right-hand column, line 21 - page 318, right-hand column, line 22; figures 1, 2 ----- -/-- </td> <td>1-23</td> </tr> </tbody> </table>			Category ^a	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	X	R. BALDONI ET AL: "Characterization of consistent global checkpoints in large-scale distributed systems", PROCEEDINGS OF THE FIFTH IEEE COMPUTER SOCIETY WORKSHOP ON FUTURE TRENDS OF DISTRIBUTED COMPUTING SYSTEMS, 1 January 1995 (1995-01-01), pages 314-323, XP55026754, DOI: 10.1109/FTDCS.1995.525000 ISBN: 978-0-81-867125-8 page 314, right-hand column, line 21 - page 318, right-hand column, line 22; figures 1, 2 ----- -/--	1-23
Category ^a	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.						
X	R. BALDONI ET AL: "Characterization of consistent global checkpoints in large-scale distributed systems", PROCEEDINGS OF THE FIFTH IEEE COMPUTER SOCIETY WORKSHOP ON FUTURE TRENDS OF DISTRIBUTED COMPUTING SYSTEMS, 1 January 1995 (1995-01-01), pages 314-323, XP55026754, DOI: 10.1109/FTDCS.1995.525000 ISBN: 978-0-81-867125-8 page 314, right-hand column, line 21 - page 318, right-hand column, line 22; figures 1, 2 ----- -/--	1-23						
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C.		<input checked="" type="checkbox"/> See patent family annex.						
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed								
Date of the actual completion of the international search 10 May 2012		Date of mailing of the international search report 22/05/2012						
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Michel, Thierry						

INTERNATIONAL SEARCH REPORT

International application No PCT/US2012/025388

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	HIGAKI H ET AL: "Checkpoint and rollback in asynchronous distributed systems", INFOCOM '97. SIXTEENTH ANNUAL JOINT CONFERENCE OF THE IEEE COMPUTER AND COMMUNICATIONS SOCIETIES. DRIVING THE INFORMATION REVOLUTION., PROCEEDINGS IEEE KOBE, JAPAN 7-11 APRIL 1997, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, US, vol. 3, 7 April 1997 (1997-04-07), pages 998-1005, XP010251940, DOI: 10.1109/INFCOM.1997.631114 ISBN: 978-0-8186-7780-9 page 999, right-hand column, line 1 - page 1002, left-hand column, line 25 ----- YUQUN CHEN ET AL: "CLIP: A Checkpointing Tool for Message Passing Parallel Programs", PROCEEDINGS OF THE 2ND ACM WORKSHOP ON ROLE-BASED ACCESS CONTROL. RBAC '97. FAIRFAX, VA, NOV. 6 - 7, 1997; [ACM ROLE-BASED ACCESS CONTROL WORKSHOP], NEW YORK, NY : ACM, US, 15 November 1997 (1997-11-15), pages 33-33, XP010893034, ISBN: 978-0-89791-985-2 page 35, left-hand column, line 5 - page 39, right-hand column, line 24 ----- Ferda Tartanoglu: "Using the B Method for the Formalization of Coordinated Atomic Actions", CS-TR-865, Technical Report Series, 1 October 2004 (2004-10-01), pages 1-18, XP55026374, University of Newcastle upon Tyne, UK Retrieved from the Internet: URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.8623&rep=rep1&type=pdf [retrieved on 2012-05-07] page 8, line 1 - page 10, line 17 ----- US 5 819 021 A (STANFILL CRAIG [US] ET AL) 6 October 1998 (1998-10-06) cited in the application the whole document -----	1-23 1,11, 14-18, 21-23 1,10,11, 18,21-23 1-23

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2012/025388

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
US 5819021	A 06-10-1998	AT 359551	T 15-05-2007	
		AU 1286197	A 17-07-1997	
		CA 2240286	A1 03-07-1997	
		DE 69637020	T2 16-08-2007	
		DK 0954781	T3 06-08-2007	
		EP 0954781	A1 10-11-1999	
		ES 2283008	T3 16-10-2007	
		JP 3207438	B2 10-09-2001	
		JP H11514116	A 30-11-1999	
		PT 954781	E 31-05-2007	
		US 5819021	A 06-10-1998	
		WO 9723826	A1 03-07-1997	

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,R0,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,IL,IN,IS,JP,KE,KG,KM,KN,KP,KR,KZ,LA,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN

(72)発明者 ホーリー, ジョセフ スケフィントン ザ サード
アメリカ合衆国, マサチューセッツ州 02478, ベルモント, ヒルクレスト ロード 11
F ターム(参考) 5B027 AA01 BB01 CC04